



Vas Vármegyei Szakképzési Centrum
Nádasy Tamás Technikum és Kollégium

SZAKDOLGOZAT

Kollégiumi Étrendkezelő

Kustor Ádám Norbert És Kovács Zalán

**Konzulens:
Balics Gábor**

2024

Nyilatkozat

Alulírott, Kustor Ádám Norbert és Kovács Zalán Flórián szakos hallgatók kijelentjük, hogy a Kollégiumi Étrend Kezelő című szakdolgozat feladat kidolgozása a saját munkánk, abban csak a megjelölt forrásokat, és a megjelölt mértékben használtuk fel, az idézés szabályainak megfelelően, a hivatkozások pontos megjelölésével.

Eredményeink saját munkán, számításokon, kutatáson, valós méréseken alapulnak, és a legjobb tudásunk szerint hitelesek.

9495 Kópháza Béke u. 50, 2024.04.19.

hallgató

9423 Ágfalva Ciklámen u. 37. 2024.04.19.

hallgató

Kivonat

Kollégiumi Étrend Kezelő (KÉK)

A szakdolgozatunkban egy meglévő problémát oldottunk meg, mégpedig a kollégiumi étkezés papíralapú rendszere és esetleges lemondások nehézségeit.

A KÉK egy nagyszerű weboldal, ahol a diákok pár kattintással elérik azt, ami eddig telefonhívásokba került. Látják a teljes heti és jövőheti menüt, amit pár kattintással le tudnak mondani, ezzel is elkerülve az esetleges pazarlást és fölösleges papírnyomtatást.

A cégnek, ami az étkezésünkkel foglalkozik is egy nagyon jó felület, mivel egyszerűbben meg tudják határozni azt, hogy mennyi étellel készüljenek.

Abstract

Dormitory Diet Manager (KÉK)

In our thesis we managed to solve an existing problem, the paper-based application for food in the student hostel and the complications of the food's cancellation.

The KÉK is a great website, where students can achieve the same things, that cost them multiple phone calls. They see the whole menu for this week and for next week, which they can cancel with only a couple of clicks. This will help to avoid food waste and unnecessary printing.

This is a really good website for the company that brings the food as well, because they can plan forward that how much food they have to bring.

Tartalom

1.	Bevezetés	9
1.1.	Ezeket használtuk az oldalunk elkészítésére.....	10
2.	Hasonló alkalmazások	11
3.	Felhasználói dokumentáció	12
3.1.	Software és Hardware igény	12
3.2.	Az oldal elérése.....	13
3.3.	A belépés és a regisztráció	13
3.4.	Az oldalak tartalma	13
3.4.1.	Heti menü oldal	13
3.4.2.	Profil oldal	14
3.4.3.	Admin oldal	14
3.5.	Diák felhasználó	14
3.5.1.	Az e heti és a jövő heti napok lemondása.....	14
3.5.2.	Profil szerkesztése	14
3.6.	Admin felhasználó	14
3.6.1.	Eheti és jövő heti menü szerkesztése.....	14
3.6.2.	Felhasználói profilok törlése	15
3.7.	FAQ vagyis gyakran felmerült kérdések.....	15
3.7.1.	Mi van, ha a felhasználó elfelejtette a kódját?.....	15
3.7.2.	Az oldalon hogyan akarjuk megoldani a későbbiekben a fizetést?	15
3.7.3.	Hogyan védjük meg a programunkat attól, hogy akárki be tudjon rá regisztrálni?	15
3.7.4.	Hogyan tudom lemondani az étkezést véglegesen?	15
3.7.5.	És a tanárok is használhatják az oldalt, ha szeretnék?.....	16
4.	Fejlesztői dokumentáció	17

4.1.	A frontend	17
4.1.1.	A komponensek	17
4.1.2.	Fontosabb package-k frontenden.....	17
4.2.	A backend	17
4.2.1.	Az adatbázis táblái.....	17
4.2.2.	Fontosabb package-k backenden	19
4.2.3.	Az admin oldal	20
5.	Az oldal tesztelése	21
5.1.	Port kezelési hiba	21
6.	Az oldal designja	22
7.	Összefoglalás	23
7.1.	Ennek a szakdolgozatnak, programnak a célja	23
7.2.	A program megvalósítása.....	23
8.	Fejlesztési lehetőségek	25
9.	Az oldal újratekzdése.....	27
10.	Felhasznált weboldalak és alkalmazások	28
10.1.	A colors nevű oldal színei	28
10.2.	A freepik nevű oldal képei	28
10.3.	A flaticon nevű oldal ikonjai	28
10.4.	A figma.....	28
10.5.	Microsoft designer.....	28
10.6.	Postman	29
11.	Összegzés	30
12.	Mellékletek és kódok.....	31
12.1.	Profil.jsx	31

12.2.	Loginform.....	32
12.3.	Navbar.jsx.....	33
13.	Irodalomjegyzék.....	35

1. Bevezetés

A 2024-es szakdolgozatunk témájának egy olyan weboldalt választottunk, ami elősegíti a diákok kollégiumi életét. Legfőbb célja az olyan lassan már elavult módszerek leváltása, mint például a papír alapú regisztráció vagy a telefonos lemondás, azt szeretnénk volna elérni, hogy mind a cégnek, tanároknak és diákoknak egyaránt megspóroljunk egy kis munkát.

Úgy gondoltuk, hogy web alapon lesz a legegyszerűbben elérhető mindenki számára, mivel nincs platformhoz és eszközhöz kötve. És a diákok otthon az ágyukból is elérhetik.

A céges oldalon látszik, hogy ki, mit és mikor mondta le, ezzel is pontosítva, hogy bizonyos napokra mennyi étellel készüljenek. Ezek mellett a cégnek van lehetősége üzenetet írni a diákok felé, ha már valamit elintézett vagy esetleg nem lesz valamilyen program miatt étel szállítva a diákoknak.

A weboldalunkon a diákok látják az egész heti és a jövő heti menüt is, amit lemondhatnak 1 nappal az étkezés előtt. Ha lemondanak egy bizonyos napot, akkor az elszürkül, ezzel is jelezve azt, hogy számára már nem elérhető. Péntekenként nincs vacsora a kollégiumban ezért az nem is látszik. Diákként látod az üzeneteket is idő szerint rendszerezve. Ezek mellett A diákoknak mind van egy profilja, ahol látják az adataikat és szükség esetén tudják változtatni is azt.

A saját tapasztalataink inspiráltak arra, hogy létrehozzuk ezt az oldalt, mivel mi is meg tapasztaltuk a jelenlegi rendszer nehézségeit. A szakdolgozatunkat tudásunkhoz mérten osztottuk fel. Zalán foglalkozott a frontend-el, míg én dolgoztam a backend-en és ketten írunk a dokumentációt.

1.1. Ezeket használtuk az oldalunk elkészítésére

- A szoftverünk frontend fejlesztéséhez React keretrendszert használtunk, backend fejlesztéséhez pedig JavaScript-et használtam.
- Ezekhez a Visual Studio Code fejlesztői környezeteket használtuk.
- A lokális környezet kialakításához Laragont használtunk.
- Az adatok tárolására MySQL szerverben történik.

2. Hasonló alkalmazások

Jelenleg nincs sok ilyen és ehhez hasonló program, oldal és szoftver Magyarországon. Ami nagyon szomorú, hiszen ezek és az ilyesfajta oldalak nagyon hasznosak tudnak lenni. Emellett az éveink alatt igenis láttuk, hogy egy ilyen oldal program vagy szoftver nagy segítség lett volna számunkra.

Az emenza nevű oldal az, amit ki tudnék emelni példaként. Az emenza ugyan úgy, mint a mi a mi oldalunk képes arra, hogy a diákok lemondják az étkezéseiket. De viszont ott a szülők is képesek lemondani a gyermekeik étkezéseiket. Ugyanis az emenza nem csak gimnáziumokat, technikumot és egyetemeket céloz, hanem óvodákat és általános iskolákat is. Vagyis mivel egy óvoda nem képes lemondani a saját étkezését, ezért a szülő el tudja intézni helyette. Fontos hozzá tennem, hogy a nem tudom, hogy az emenza mennyire egy egyszerű és letisztult oldal. De mi mindenféleképpen arra törekedtünk, hogy nagyon egyszerűen kezelni lehessen az oldalt és átlátható legyen minden.

3. Felhasználói dokumentáció

Most hadd beszéljek kicsit arról, hogy a felhasználóknak miket tudnak csinálni az oldalon.

Átlagos felhasználók, azaz diákok lehetőségei az oldalon:

- A felhasználók látják a teljes menüt két hétre.
- A felhasználó le tudja mondani ezeket az étkezéseket.

Különleges felhasználók, azaz adminok lehetőségei az oldalon:

- Az eheti és a jövő heti étrend szerkesztése és a napot.
- Diákok profiljainak a szerkesztése.
- Diákok és lemondások listázása
- Az adminok képesek a diákoknak profilokat létrehozni.
- Az adminok képesek megváltoztatni a menüket és a napot.

3.1. Software és Hardware igény

Az oldal, vagy program futtatásához nem is kell más, mint egy böngésző és hálózati kapcsolat. Vagyis nem kell semmit sem telepíteni a gépünkre.

Ezek a specifikációk ajánlottak az oldalhoz:

- 1 GB szabad memória
- Windows 7 de inkább újabb operációs rendszerek, mint például a Windows 10 vagy a Windows 11
- A processzor legalább legyen 1 GHz-es

3.2. Az oldal elérése

Az oldalunkat, mint már említett interneten keresztül lehet elérni egy szinte akármilyen böngészőből.

Erre tökéletes példák:

- Google Chrome
- Opera
- Firefox

3.3. A belépés és a regisztráció

Fontos kiemelnem, hogy az oldalt nem tudod használni, ha nincsen profilod.

Vagyis mielőtt elkezded használni az oldalt, azelőtt kell szerezned egy saját profilt.

Profilt a diákok viszont nem képesek regisztrálni. Ugyanis az adminok fogják létrehozni a diákok profiljait. A regisztrációnál meg kell adnunk az adminoknak a teljes nevünket, az email címünket, a telefonszámunkat, a jelszónkat és a diákigazolvány számunkat.

Belépésnél pedig csak az felhasználónevünkre és a jelszónkra van szükségünk.

3.4. Az oldalak tartalma

3.4.1. Heti menü oldal

Az oldalunkon két menüpont van a heti menük elérésére. Az „Ez a hét” oldalon találhatjuk meg az eheti menüt, a „Jövő hét” menüpontnál pedig a jövő hetit. Ezeken az oldalakon tudjuk gombokkal lemondani azokat a napokat, amelyeken nem szeretnénk étkezni. Itt is az oldal tetején van a navigációs menü és az oldal közepén található 5 doboz az 5 nappal. A dobozokon belül természetesen látjuk a napokat. A navbar-ban található az e-heti és a jövőheti menüknek az elérése.

3.4.2. Profil oldal

Ezt az oldalt is elérik az adminok és a diákok is. Ezen az oldalon találjuk meg a saját profilunknak az adatait a jelszón kívül. Ebből kifolyólag a jelszót ezen az oldalon nem képesek a felhasználók átírni. Ha valamilyen adatot átírunk, akkor a frissítés gombra kell kattintani annak az elmentéséhez.

3.4.3. Admin oldal

Az admin oldalon számos dolgot tudunk megtenni. Ezen az oldalon láthatjuk az összes felhasználót az adataikkal együtt. Ezeket az adatokat az admin felhasználó meg tudja változtatni. Emellett fel tud venni új felhasználót, látjuk az összes lemondást és a lemondásokat tudja törölni az admin felhasználó. Látjuk a teljes menüt mind a két hétre, a menüt itt lehet megváltoztatni.

3.5. Diák felhasználó

3.5.1. Az e heti és a jövő heti napok lemondása

A diákok képesek lemondani a menü nevű menüpont alatt azokat a napokat, amiken nem szeretnék enni. Ezt az oldalon lévő napokhoz rendelt gombok segítségével tudják megtenni gyorsan és egyszerűen. A mentés gombbal pedig elmentik a változtatásokat.

3.5.2. Profil szerkesztése

A diákok képesek a profiljaikat szerkeszteni a profil menüpont alatt. Ebbe bele tartozik a saját nevük, a telefonszámuk és a diákigazolvány számuk. A frissítés gombra kattintva fel is frissülnek az adataik és elmentésre kerülnek.

3.6. Admin felhasználó

3.6.1. Eheti és jövő heti menü szerkesztése

Az admin oldalon az admin felhasználók képesek az e heti és a jövő heti menüket szerkeszteni. Azon belül minden napot tudnak szerkeszteni, és azon belül a reggelit, ebédet és a vacsorát. Ezt a mentés gombbal képesek elmenteni.

3.6.2. Felhasználói profilok törlése

Az adminok az admin oldalon képesek kitörölni a diákok profiljait.

3.7. FAQ vagyis gyakran felmerült kérdések

3.7.1. Mi van, ha a felhasználó elfelejtette a kódját?

Ezen sokat gondolkodtunk, hogy ilyenkor mi legyen. Két féle megoldásra gondoltunk, amit a jövőben meg szeretnénk valósítani. Az első az az, hogy a bejelentkezés oldalon lenne egy „Elfelejtett jelszó” amire rákattintva a felhasználó a regisztrált emailre tudna küldeni egy jelszó helyreállító linket, ami elvinne egy oldalra, ahol a felhasználó át tudná írni a jelszavát. A másik az az, hogy a felhasználó felvehetné az adminokkal a kapcsolatot, akik az admin oldalon át tudnák írni a jelszavát.

3.7.2. Az oldalon hogyan akarjuk megoldani a későbbiekben a fizetést?

Voltak ötleteink, hogy ezt hogyan oldanánk meg ha kellene nekünk ez az oldalra. Ahogyan most áll az egész projekt, az az ötletünk, hogy a fizetni egyelőre az iskolának kell és majd az iskola eljuttatja az összeget hozzánk. Erre mindig a hónap végén esne sor, Természetesen gondoltunk más megoldásra is. A másik ötlet az az volt, hogy kirakunk az oldalra egy számlaszámot, ahova hónap végén a diáknak el kell küldenie azt az összeget, amennyiért evett. Természetesen azokat a napokat amiken nem ettek a diákok, azokat a napokat nem kell, hogy kifizessék.

3.7.3. Hogyan védjük meg a programunkat attól, hogy akárki be tudjon rá regisztrálni?

Természetesen ezt el szeretnénk kerülni, hogy akárki be tudjon regisztrálni az oldalra. Ezt úgy képzeljük el, hogy a diákigazolvány számát a diáknak be kell írnia a regisztrációnál és mi azt le fogjuk ellenőrizni. Ez egy nagyon egyszerű és praktikus megoldás, de mindenféleképpen kiszűri az embereket.

3.7.4. Hogyan tudom lemondani az étkezést véglegesen?

Az étkezést véglegesen lemondani csak az adminokkal való kapcsolatba lépéssel lehet. De a későbbiekben lesz rá lehetőség, hogy az oldalon keresztül le lehessen mondani pár kattintással. Természetesen ez csak a következő hónaptól kerül lemondásra.

3.7.5. És a tanárok is használhatják az oldalt, ha szeretnék?

A válasz igen. Nem tanároknak lett az oldal tervezve persze. Csak diákoknak. De ha egy tanár megkér minket, hogy hadd használja ő is a szolgáltatásunkat, akkor létrehozunk neki külön egy diák profilt, ugyanis a nap végén a tanár és a diák profil ugyan azt tudná. Viszont azért hozzuk mi létre a tanárnak a profilját, mert a tanárok nem rendelkeznek diákigazolvánnyal. Ebből kifolyólag ők maguk nem tudnak regisztrálni.

4. Fejlesztői dokumentáció

4.1. A frontend

4.1.1. A komponensek

4.1.1.1. A navbar

Ez a components/navbar mappán belül található. Ez egy tsx fájl. Ráadásul megtalálász mellette egy css fájlt is. A css fájlban van persze annak a dizájnja.

4.1.2. Fontosabb package-k frontenden

- React-bootstrap: A React Bootstrap újraépíti a Bootstrap Javascript részét, és minden komponenst úgy hoz létre, hogy valódi React komponensként működjenek, anélkül, hogy szükségtelen kódot tartalmazzanak. Ez lehetővé teszi a könnyű integrációt és emellett azt is, hogy a Bootstrap stílusait és komponenseit használjuk React projektben. A számunkra hasznos komponensek a Col, FormGroup, Row, Form és a Button volt.
- React: A react egy JavaScript könyvtár. Ez lehetővé teszi nekünk fejlesztők számára, hogy hatékonyan és rugalmasan építsünk felhasználói felületeket, amelyeket egymásba ágyazva és kombinálva lehet létrehozni. Ezt nagyon széles körben használják a világon a programozók.

4.2. A backend

4.2.1. Az adatbázis táblái

4.2.1.1. Roles tábla

Mezőnév	Típus
id	INT
name	VARCHAR(10)

4.2.1.2. Userrole tábla

Mezőnév	Típus
userid	INT
roleid	INT

4.2.1.3. Users tábla

Mezőnév	Típus
id	INT
name	VARCHAR(45)
password	VARCHAR(70)
email	VARCHAR(45)
phone	VARCHAR(12)
p_name	VARCHAR(45)
p_phone	VARCHAR(12)
role_id	INT

4.2.1.4. Cancelation tábla

Mezőnév	Típus
id	INT
student_id	INT
weeklyspecially_id	INT

4.2.1.5. *Weeklyspecial tábla*

Mezőnév	Típus
id	INT
breakfast	VARCHAR(45)
lunch	VARCHAR(45)
dinner	VARCHAR(45)
day	DateTime

4.2.1.6. *Messages tábla*

Mezőnév	Típus
id	INT
messages	VARCHAR(200)
time	DATETIME

4.2.2. **Fontosabb package-k backenden**

- Cors: A Cors egy mechanizmus a webfejlesztésben, amely lehetővé teszi a biztonságos adatátvitelt a böngészők és a szerverek között. A Cors segítségével a webalkalmazások képesek kommunikálni más erőforrásokkal, amelyek különböző eredetből származnak. Itt gondolok domainre vagy portra. Emellett a Cors nagyon fontos a webalkalmazások biztonsága végett, mivel ez megakadályozza a rosszindulatú kódok által végrehajtott támadásokat.
- Jsonwebtoken: A Jsonwebtoken lehetővé teszi az adatok biztonságos átvitelét két fél között. Ezt az fejlesztők gyakran használják autentikáció és autorizáció során. Amikor a felhasználó bejelentkezik az alkalmazásba, akkor a szerver létrehoz egy Jsonwebtoken-t, amiben benne van, vagyis tartalmazza a felhasználó azonosítóját és más fontos információkat. Ezután a Jsonwebtoken klienshez hűldi, és a kliens minden további kérdésben elküldi a szervernek a Jsonwebtoken-t.
- Express: Az Express egy Node.js webalkalmazás keretrendszer. Az express

rugalmas és könnyű útvonalvezetési keretrendszert biztosít minimális alapfunkciókkal, amelyeket ki lehet egészíteni. Az Express segítségével könnyedén létrehozhatunk API-kat és webalkalmazásokat.

- Axios: Ez egy http klienskönyvtár. Ez lehetővé teszi nekünk fejlesztőknek, hogy egyszerűen, könnyedén és hatékonyan HTTP kéréseket hajtsunk végre. Emellett lehetőséget nyújt kérések és válaszok közbeavatkozására, ráadásul a kérések transzformálására is. Ez egy nagyon népszerű választás a webes alkalmazások fejlesztéséhez, amelyek kommunikálnak egy szerverrel.

4.2.3. Az admin oldal

Az admin oldalon rengeteg dolgot tud csinálni az admin. Először is meg tudja változtatni akármelyik diáknak a kódját, meg tudja változtatni akármelyik diáknak az e-mail címét, a telefonszámát és még akár a diák szülője nevét is, emellett a szülője telefonszámát. De emellett meg tudja változtatni a szülő telefonszámát is. Természetesen ezekből az adminok újakat is létre tudnak hozni. Mint említettem a diák felhasználókat az adminoknak kell, hogy létrehozzák. Ebből kifolyólag ezen az oldalon van az „Új felhasználó hozzáadása” részleg. Ráadásul itt az adminok tudják törölni a diákokat is. És végül az adminok ezen az oldalon képesek megváltoztatni a menüket.

5. Az oldal tesztelése

Én magam teszteltem az oldalt a fejlesztéssel párhuzamosan, emellett miután végeztem a fejlesztéssel többször is leteszteltem. Az én saját gépemen vittem véghez a tesztelést, aminek a lényeges specifikációi: 16GB RAM és i7-11800H

A program futása mindenféle hiba nélkül történt és a memória használata is alacsony maradt.

Leteszteltettem az oldalt mással is. Azt mondta, hogy az oldal szerinte letisztult és nagyon egyszerű, lényegre törő és felhasználóbarát volt. Könnyen tájékozódott az oldalon és nem futott hibákba.

De persze természetesen, ha a végső teszteléseken is voltak hibák, a fejlesztéssel párhuzamos teszteléseken előfordultak hibák amik persze ki lettek javítva. Szerencséből nagy hibába nem ütköztünk.

Később megpróbáltam más gépeken is az oldalt tesztelni. Egy 8gb ramos i5-ös processzorral rendelkező számítógépen is leteszteltem, és azon is zökkenőmentesen futott.

5.1. Port kezelési hiba

Az Ádámnál felmerült egy hiba, ami egy jó ideig lekötötte őt. Az volt a probléma, hogy a laptopja valamiért folyamatosan letiltotta a portot 1 kérés után. Ez azt eredményezte, hogy egy ideig nem tudott haladni a munkájával. Kellet azért idő mire kiderült az, hogy ez állt a hibájának a hátterében.

6. Az oldal designja

Az oldalt már régen elképzeltük a fejlesztés előtt, hogy hogyan szeretnénk, hogy kinézzon.

Az oldal kék színe nem az oldal nevéből adódott, ugyanis már jóval az oldal neve előtt tudtuk, hogy az oldal színe kék kell, hogy legyen. Tehát az oldal színe mindenféleképpen többet hozzá ad az oldal nevéhez. (Kollégiumi Étrendkezelő vagyis Kék) Így utal arra amire az oldal használva van és annak kék színére. Azt szerettük volna, hogy az oldalon legyenek képek és matricák az oldalhoz kapcsolódóan, vagyis a képek és a matricáknak legyen köze az ételekhez.. A felhasznált ikonokat és képeket az internetről töltöttem le és azokat akár meg is szerkesztettem, vagy egyszerűen csak paintbe, vagy akár photoshopban is. Az oldalt a lehető legfelhasználóbarát típusúra szerettük volna megcsinálni. Ez volt az egyik legelső gondolat, ami megfordult a fejünkben.

Hiszen mindenki szereti a lényegretörő, egyszerű oldalakat.

7. Összefoglalás

7.1. Ennek a szakdolgozatnak, programnak a célja

Tehát, mi is volt a cél? A célunk egy olyan program volt, amivel a diákok könnyű módon képesek a napi étkezéseiket lemondani és meg is nézni azt. Egy ilyen oldalra azért van szükség mert ez így sokkal egyszerűbb megoldás, mint betelefonálgatni az étel lemondásáért. Ez azért jutott eszünkbe, mert minekünk nagyon kellett volna egy ilyen program az évek alatt. Sokkal egyszerűbb lett volna a helyzet. Tehát ez az oldal a saját frusztrációnkból készült.

7.2. A program megvalósítása

Először átgondoltuk azt, hogy mit is szeretnénk az oldalon látni és mit szeretnénk, hogy a program képes legyen csinálni és papírra vetettük azt. Elgondolkodtunk azon, hogy körülbelül hogyan nézzen ki az oldal és hogy merre legyenek az oldalon elhelyezve a dolgok, hogy a lehető legegyszerűbben meg lehessen őket találni. Ebből kifolyólag készült egy pár kisebb rajz, hogy hogyan szeretnénk az oldalt. Ezután a figma nevű oldalt felhasználva megcsináltuk a megbeszéltek és a kis rajzok alapján az oldal kinézetét, hogy szemmel is láthassuk, hogy mit szeretnénk. Miután kész volt a figma-s design, azután megmutattuk másoknak, hogy mit gondolnak az oldalról. Akiknek mutattuk azoktól pozitív visszajelzéseket kaptunk nagy örömünkre. Aztán elgondolkodtunk azon, hogy ezt hogy tudnák megvalósítani.

Ezután eldöntöttük, hogy ki melyik részét fogja megcsinálni a programnak. Ádám csinálta az oldal backendjét, amibe beletartozott az adatbázis is, Zalán pedig az oldal frontendjének állt neki. A program fejlesztése alatt természetesen rengeteg ötletet és ihletet kaptunk, és emellett voltak olyan ötletek is, amiket inkább hagytunk. Nem azért természetesen mert nem tudtuk volna megcsinálni, hanem azért, mert volt olyan ötlet, amit a valóságban nem tudtak volna kivitelezni emberek. De voltak olyan ötleteink is, amiket rájöttünk arra, hogy nagyon feleslegesek lennének az oldalon. Jó persze volt olyan is, amit lehet nem tudtunk volna megcsinálni, de akkor is inkább megpróbáltuk, minthogy egyből elvessük az ötletet. Vagyis a program készítése közben voltak próbálkozások ezekkel a hirtelen jött ötletekkel. Emiatt azért voltak nagyobb és kisebb változtatások egyaránt a

fejlesztés alatt.

Zalán először a főoldalnak állt neki a figma alapján és azután megcsinálta a navbar-t (ez a navbar később ki lett cserélve, mert Ádám talált egy jobb megoldást annak a megvalósítására), hogy át lehessen lépni az oldal többi lapjára. Ezután megcsinálta a bejelentkezést és a regisztrációt, hogy amikor az ember belép az oldalra akkor egyből a bejelentkezésnél találja magát. Azután neki állt menü oldalának, ahol nagyon sok újra tervezés volt arról, hogy ezt, hogy lehetne úgy megcsinálni, hogy egyszerű és átlátható legyen. Ugyanis áttért a profil oldal programozására. Itt először még nem volt tiszta, hogy mit lehessen szerkeszteni és mit nem. De ez idővel, ahogyan a program készült kialakult. Ezután lett a sidebarba egy kijelentkezés gomb. Az alatt pedig megtalálható az adminok oldala. Az adminok oldala jó sok ideig hagyva lett, ugyanis még egy jó ideig nem tudtuk, hogy mekkora hatalmat szeretnénk adni az admin profilnak. Emiatt ez is később, miután a legtöbb dolog már kész lett, azután lett megcsinálva.

Ádám neki is állt az oldal adatbázisnak és a backendnek, miután mindent megbeszéltünk. A backendhez a Visual Studio 2022 nevű programot használta. De mindenekelőtt megpróbálta kitalálni, hogy hogyan lehetne az adatbázis tábláit a legjobb módon összekapcsolni. Emellett létrehozta adatbázist mely eltárolja az adatainkat. Ez az adatbázis KÉK_DB nevet kapta.

A legvégén pedig összekötöttük az adatbázist és mellette a backendet a fronteddel.

8. Fejlesztési lehetőségek

Az oldalt elég könnyen lehet a továbbiakban fejleszteni. Emiatt eléggé sok ötletünk van amivel még ki tudnánk bővíteni a jövőben az oldalt.

- ✚ Elgondolkodtunk azon, hogy a jövőben ne csak a napokat lehessen lemondani, hanem csak bizonyos étkezéseket. Vagyis külön le lehessen mondani a reggelit, az ebédet és a vacsorát.
- ✚ Először úgy szeretnénk volna, hogy lehessen az oldalon választani A és B menü között. Ez azért nem valósult meg mert sok helyen biztos nem szeretnék 2 fajta ételt elkészíteni. Ami szomorú, hiszen ez egy remek ötlet lenne. De a jövőben lehet, hogy ezt is megcsináljuk.
- ✚ Ha egy diák felhasználó elfelejti a jelszavát, akkor a bejelentkezés oldalon képes legyen jelszó helyreállítót kérni az e-mailjára. Vagy a diák megkérhetné az adminokat hogy írják át a jelszavát. Az adminok ez azt admin oldalon tudnák megtenni.
- ✚ A későbbiekben mindenféleképpen meg kéne csinálni azt, hogy a diák az emailját is meg tudja változtatni a profil oldalon.
- ✚ Volt egy olyan ötletünk, hogy belerakjuk az oldalba a snake nevű játékot. Erre a kis poénes dologra azért gondoltunk, hogy a diák ha a sorban áll az ételért akkor tudjon snakezni. Ezt a funkciót „Hosszú a sor?”-nak neveztük el. Ez egy ötlet volt amibe bele is lett kezdve később, de idő hiányába ezt sajnos meg kellett szüntetni.
- ✚ Azt is szeretnénk, hogy az összes eszközön (itt gondolok például a mobilokra és a tabletekre) elérhető legyen ez a program. Ezt a jövőben biztos, hogy meg szeretnénk csinálni.
- ✚ Az oldalon mindenféleképpen a későbbiekben lesz az admin oldalnak egy része, ahol látjuk azt, hogy melyik diák hány napot mondott le a hónapban. Ez azért lesz fontos, ugyanis ennek segítségével a hónap végén ne keljen a diákoknak kifizetniük azokat a napokat, amiket lemondtak.
- ✚ Szeretnénk egy olyan részt az oldalon, ahol az adminoknak a diákok képesek üzeneteket küldeni, ha valamilyen probléma merülne fel.
- ✚ Nagyon egyszerű és hasznos lenne egy mód az oldalon, amivel a diákok le tudnák mondani az étkezésüket véglegesen. Emiatt majd csinálni szeretnénk egy végleges étkezés lemondó részt az oldalon.
- ✚ A program dizájnját majd még szeretnénk alakítani, fejleszteni.

- 🌈 Egy olyan oldal, ahol az adminok a diákoknak képesek az étkezéssel kapcsolatban kiírni üzeneteket.

Ezek alapján kimondhatom, hogy az oldal még bőven fejleszthető. Rengeteg a lehetőség, amit ki szeretnénk használni. Persze ettől függetlenül ezeket is úgy szeretnénk majd megcsinálni, hogy az oldal maradjon egyszerűen kezelhető és átlátható. Természetesen azért is szeretnénk ezeket megcsinálni, hogy tanuljunk belőle és fejlesszük magunkat. És persze azért mert az oldalunkat egyszer újra kezdtük, emiatt még jobban szeretnénk a második próbálkozásból egy jobb oldalt kihozni.

9. Az oldal újratekzdése

Volt egy pont a programozásunk ideje alatt, amikor is úgy lett döntve, hogy valamilyen szinten újra kezdjük az oldalt. Ez azért kellett, mert az oldalknak a kódja nem volt a legszebben megírva, és sok voltak dolgok amiket meg akartunk alapjáratat változtatni. Azt szerettük volna, hogy a kódokat is egyszerűen át lehessen látni. Ráadásul a program úgy lett megcsinálva, hogy valamiért egyszerűen sok dolog a backendnél nem akart működni. A háttérben nagyon sok hibát jelzett az oldal, amiket sok idő lett volna kijavítani, vagy már csak rájönni arra, hogy mi lehet a probléma. És ezek mellett emiatt úgy lett elhatározva, hogy újra lesz kezdve valamennyire. Még ha nem is lesz szép a fronted design-je akkor is inkább működjön az oldal backendje. Miután újra kezdtük több volt a tudásunk a több hónapnyi programozás után ezen a téren, ebből kifolyólag a kódokat sokkal jobban, átláthatóbban és hiba mentesen meg tudtuk írni. Ráadásul sokkal egyszerűbben össze tudtuk kötni az oldalt az adatbázissal. Úgy gondolom a döntésünk nem volt egy rossz döntés. Ugyanis ebből az oldalból egy sokkal jobb oldalt lehet kihozni hosszútávon, mint az eredetivel. A design-t is kicsit megváltoztattuk, a kék szín természetesen megmaradt. Ezen az oldalon már el tudtuk érni, hogy adatokat tudjunk bevinni, törölni és módosítani az adatbázisunkban. Ami egy nagyszerű dolog volt. Ezzel az oldallal azon a rövid időn belül, amin dolgoztunk rajta, sokkal egyszerűbb volt már minden, mert sokkal precízebben oldottuk meg. Természetesen pár ötletet ki kellett dobunk sajnos emiatt, mert már nem volt elég időnk. De úgy gondolom, hogy így sokkal jobb, hogy van egy oldal, ami lehet, hogy még nem néz ki a legszebben, akkor is legalább működik a backend és az adatbázis. Az első oldalba inkább több design-olás volt, mint backend.

10. Felhasznált weboldalak és alkalmazások

10.1. A colors nevű oldal színei

<https://colors.co/>

Ez az oldal segített nekünk a kéknek az árnyalatainak a kiválasztásában. Ez az oldal szín sémákat, szín kombinációkat mutat. Ezzel segítve a programozók életét, ugyanis nagyszerű témákat és szín kombinációkat lehet találni az oldalainkhoz.

10.2. A freepik nevű oldal képei

<https://www.freepik.com/>

A freepik-nek köszönhetjük az oldalunkon lévő képeket. Itt nagyon sok kép van feltöltve, amit akárki akármikor ingyen használhat akárhol. Mint mi az oldalunkon.

10.3. A flaticon nevű oldal ikonjai

<https://www.flaticon.com/>

A flaticon nevű oldalról szereztük be a logónkat. Ezen az oldalon az emberek ingyenes logókat és matricákat tudnak maguknak keresni.

10.4. A figma

<https://www.figma.com/>

Ez az oldal segített nekünk rengeteget az elején. Ugyanis nagyon jó volt megcsinálni rajta az oldalunkat, hogy lássuk, hogy milyen oldalt szeretnénk a végére. Ezen az oldalon megcsinálhatod a saját oldalad kinézetét, hogy ne csak egy kép legyen a fejedben arról, hogy milyen oldalt akarsz.

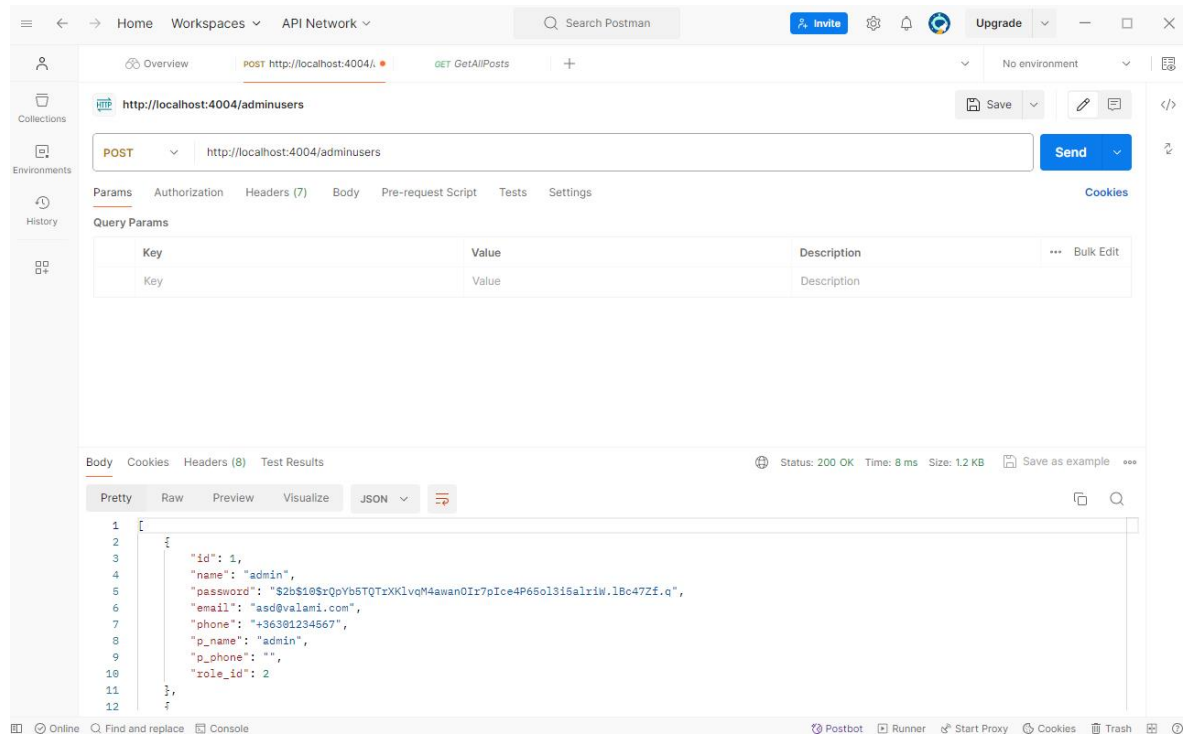
10.5. Microsoft designer

<https://designer.microsoft.com/image-creator>

Ennek az oldalnak a segítségével generáltuk le a loginunk hátterét.

10.6. Postman

A teszteléshez a Postman alkalmazást használtunk, ami az API-k tesztelésében segít.



11. Összegzés

Tehát így összegzésként elmondhatom, hogyha még a weboldalunk most nem is néz ki a legszebben, attól függetlenül a backendje és az adatbázisa nagyon jó. A nagy újra kezdés előtt volt design-je az oldalnak, de fel kellett áldoznunk azért, hogy sokkal jobban meg legyen kódolva. De ez még nem is akkora probléma, mert a weboldal design-jét a jövőben tervezzük megcsinálni és miután már egyet megcsináltunk, másodjára sokkal könnyebb lesz. A React Bootstrap egy iszonyúan jó döntés volt. Sokkal szórakoztatóbb és egyszerűbb volt a React Bootstrap mint gondoltuk volna és sokat tanultunk ezek alatt a hónapok alatt. Tömérdek dolgot bele lehet importálni emellett, ami szintűgy segít nekünk programozóknak.

Az oldalon tehát még sok dolgot frissíteni szeretnénk. Például üzenetek oldalt, ahova az adminok posztolni és üzeneteket, híreket tudnak kitenni a diákoknak vagy azt, hogy a diákok képesek legyenek az oldalon keresztül véglegesen lemondani az étkezést. Ezalatt azt értem, hogy rengeteg ötletünk van, és majd mindenféleképpen meg szeretnénk őket valósítani. Természetesen ezek a frissítések mind a diákokat és mind az adminokat érinti. Mind a két fajta profil kapni fog újdonásokat a weboldalon. És a profilokat még nem képesek a diákok. Azt majd a jövőben, de addig is az admin létrehozza nekik a saját profiljukat. Addig is viszont az adminok képesek adatot a diákok profiljait szerkeszteni, törölni, létrehozni. Azon belül pedig, az e-mail-t, a nevüket, a telefonszámukat, a szüleik nevét, a szüleik telefonszámát és a kódjukat. Emellett tudják szerkeszteni a heti menüket is.

A régi oldal végleges tesztelése alatt voltak hibák a backendben, de az új oldal gördülékenyen működik. Persze mi folyamatosan teszteltük mind a kétszer a weboldalakat, és a régi backendje akkor sem volt a legjobb, de az újjal még nem voltak problémák. Viszont egyszer volt egy nagyobb hiba, amikor az Ádámnak port kezelési hibája volt és azt napokig próbálta helyre rakni. Mint már említettem a második oldal is színe is kék maradt. Ezt muszáj volt benne hagynunk, mert nekünk tetszik, hogy az oldal neve utal az oldal design-ára is. Amellett persze, hogy az oldal teljes neve Kollégiumi Étrendkezelő.

Nagyon örülünk mellesleg, hogy ilyen oldalak léteznek, mint a flaticon, a coolers, a figma és a freepik. Ugyanis Ezek az oldalak nagyon sokat segítettek nekünk mikor képek és színek kellettek. A figma pedig óriási segítség volt abban, hogy megcsinálhattuk az oldalunk kinézetét, hogy ne csak elképzelni tudjuk az oldalt.

12. Mellékletek és kódok

12.1. Profil.jsx

```
export default function Profil() {

  return (

    <div classname="poss">

      <Row>
        <FormGroup>

          <FormGroup>
            <div classname="d-flex" style={{ height: '100px', overflow:
'scroll initial' }}> </div>

          </FormGroup>

        </FormGroup>

        <Col xs={12} md={6} lg={4}>
          <NavBar></NavBar>
          <form classname="form">
            <FormGroup>
              <Form.Label>Teljes Név</Form.Label>
              <Form.Control type="text" name="text"/>
            </FormGroup>
            <FormGroup>
              <Form.Label>Email cím</Form.Label>
              <Form.Control type="email" name="Email"/>
            </FormGroup>
            <FormGroup>
              <Form.Label>Telefonszám</Form.Label>
              <Form.Control type="phone" name="phonenumber"/>
            </FormGroup>
            <FormGroup>
              <Form.Label>Szülő neve</Form.Label>
              <Form.Control type="text" name="Parent name"/>
            </FormGroup>
            <FormGroup>
              <Form.Label>Szülő telefonszáma</Form.Label>
              <Form.Control type="phone" name="text"/>
            </FormGroup>

          </form>
        </Col>
      </Row>
    </div>
  );
}
```

```

        <Button variant="primary" type="submit">
            Elküldés
        </Button>
    </form>
</Col>
</Row>
</div>
)
}

```

12.2. Loginform

```

export default function LoginForm() {
    const [userid, setUserid] = useState(-1);
    const [username, setUsername] = useState('');
    const [password, setPassword] = useState('');

    function handleSubmit(event) {
        event.preventDefault();

        console.log(username, password);
        axios.post('http://127.0.0.1:4003/loginusers', {
            username: username,
            password: password
        })
        .then(res => {
            localStorage.setItem('token', res.data.token);
            try {
                const tokenParts = res.data.token.split('.');
                const payload = JSON.parse(atob(tokenParts[1]));

                setUserid(payload);
                console.log(payload);

                //window.location.href = payload.role_id === 1 ?
                'http://localhost:3000/home': "http://localhost:3000/adminpage"

            } catch (error) {
                console.error('Error decoding token:', error);
            }
        })
    }
}

```



```

    .catch(err => console.log(err));
  }

  return (
    <div className='container'>
      <form className="form" onSubmit={handleSubmit}>
        <Form.Group className="mb-3">
          <Form.Label>Felhasználónév</Form.Label>
          <Form.Control type="text" value={username} required onChange={e
=> setText(e.target.value)} />
        </Form.Group>
        <Form.Group className="mb-3">
          <Form.Label>Jelszó</Form.Label>
          <Form.Control type="password" required onChange={e =>
setPassword(e.target.value)} />
        </Form.Group>
        <Form.Group>
          <Button type="submit" variant="primary">Bejelentkezés</Button>
        </Form.Group>
      </form>
    </div>
  )
}

```

12.3. Navbar.jsx

```

function NavBar() {
  return (
    <Navbar bg="blue" variant="blue">
      <Navbar.Brand href="/profil">Profil</Navbar.Brand>
      <Nav className="mr-auto">
        <li className='blue'>KÉK</li>
        <li>
          <Nav.Link className='menupoint' href="/thisweek">Menü</Nav.Link>
        </li>
        <li>
          <Nav.Link className='menupoint2' href="/">Kijelentkezés</Nav.Link>
        </li>
      </Nav>
    </Navbar>
  )
}

```

```
      </Nav>
    </Navbar>

  );
}

export default NavBar;
```

13. Irodalomjegyzék

[A felhasznált szakirodalom megadása]

[A szerzők nevét mindenütt “Családnév, X.” formában kell megadni, ahol X. a szerző keresztnévének (keresztneveinek) kezdőbetűje. Magyar cikk esetén a vessző a családnév és a keresztnév kezdőbetűje közt elhagyható. Ha az egyértelműség megkívánja, a keresztnév kiírható teljesen is.

A dolgozat szerzője szabadon választhat, az A vagy a B típust használja.]

[A-típus:

A cikkekre való hivatkozás egy []-be írt sorszámmal történik. A sorszámozást folytonosan kell megtenni, a sorba rendezés alapja az első szerző családnéve.]

[1] Szerző1 (, Szerző2 ...): *Cikk címe*

Folyóirat neve, sorszám, kezdőoldal-végoldal, év.

[2] Szerző1 (, Szerző2 ...): *Konferencia-kiadvány-beli cikk címe*

„Konferenciakiadvány:” Konferencia neve, hely, kezdőoldal-végoldal, év.

[3] Szerző1 (, Szerző2 ...): *Könyvcím*

„Könyv:” Kiadó, hely, oldalszám, év.

[4] Szerző1 (, Szerző2 ...): *Kutatási jelentés címe* (csak publikus elérhető jelentés!)

„Kutatási jelentés”: Kutatási projekt neve, intézet, oldal, év.

[5] Szerző: *Disszertáció címe*

„PhD/kandidátusi/stb. disszertáció”: Egyetem, kar neve, év.

[6] *Internetes oldal elnevezése:*

URL, letöltés ideje (csak konkrét dokumentumra mutató URL adható meg!)

[B-típus:

A hivatkozás a szerzők családi nevéből és a kiadás évéből képezett azonosítóval történik. Kettőnél több szerző esetén az „et.al.” rövidítés használható. Pl. „[Vijayasundaram, 1986.]”, „[Meister and Sonar, 1998.]”, „[Felcman et.al., 1994.]”

Abban a ritka esetben, amikor több cikknek is azonos azonosító jutna (megegyeznek a szerzők és a kiadás éve) az év után az „a”, „b”, „c”, stb. betűk csatolandók, pl. „[Stone and Norman, 1993a.]”.

A sorba rendezés alapja a szerzők családneve, végül az év.

Ezt leszámítva a formátum ugyanaz, mint az A-típus esetén, de ekkor a hivatkozási sorszám leghagyható.]

